How transferable are the datasets collected by active learners?

David Lowell[1], Zachary C. Lipton[2], Byron C. Wallace[1]
Northeastern University[1]
Carnegie Mellon University[2]
lowell.d@husky.neu.edu, zlipton@cmu.edu,
b.wallace@northeastern.edu

Abstract

Active learning is a widely-used training strategy for maximizing predictive performance subject to a fixed annotation budget. Between rounds of training, an active learner iteratively selects examples for annotation, typically based on some measure of the model's *uncertainty*, coupling the acquired dataset with the underlying model. However, owing to the high cost of annotation and the rapid pace of model development, labeled datasets may remain valuable long after a particular model is surpassed by new technology. In this paper, we investigate the transferability of datasets collected with an *acquisition model* A to a distinct *successor model* S. We seek to characterize whether the benefits of active learning persist when A and S are different models. To this end, we consider two standard NLP tasks and associated datasets: text classification and sequence tagging. We find that training S on a dataset actively acquired with a (different) model A typically yields worse performance than when S is trained with "native" data (i.e., acquired actively using S), and often performs worse than training on i.i.d. sampled data. These findings have implications for the use of active learning in practice, suggesting that it is better suited to cases where models are updated no more frequently than labeled data.

1 Introduction

Modern machine learning systems tend to require large amounts of labeled data to work well. In particular, while deep learning has rapidly advanced the state-of-the-art results on a number of supervised learning tasks [12, 1], realizing these gains depends on large annotated datasets [24]. This is problematic because labeled data can be expensive to collect. Several lines of research have explored mechanisms to reduce the amount of supervision required to achieve acceptable predictive performance, including semi-supervised, transfer, and active learning, which we focus on here.

In *Active Learning* (AL) [3, 23], rather than consuming a set of given annotations, the learner engages the annotator in a cycle of learning, iteratively selecting training data for annotation and updating its model. *Pool-based* AL (the variant we consider) proceeds in rounds. In each round, the learner applies a scoring heuristic to examples in a pool of unlabeled instances, selecting those instances with the highest scores for labeling.[1] Because we often have access to far more unlabeled data than we can afford to annotate, it is hoped that by selecting especially informative examples, the active learner might achieve greater predictive performance than it would by

[1] This may be done either deterministically, by selecting the top-k instances, or stochastically, selecting instances with probabilities proportional to heuristic scores.

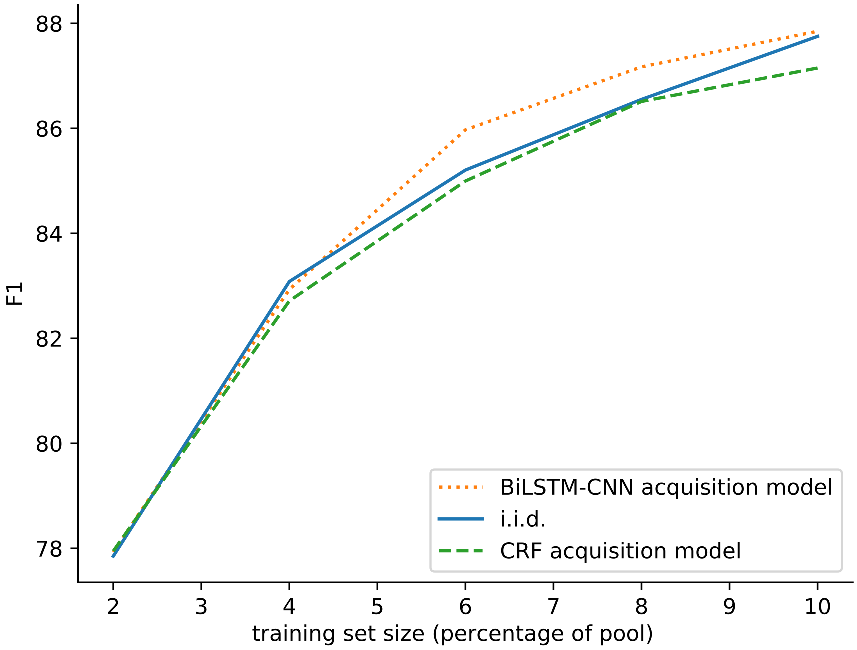


Figure 1: An example learning curve for a BiLSTM-CNN model trained to perform Named Entity Recognition (NER) using the CoNLL dataset. While active learning (yellow, dotted line) outperforms training on i.i.d. data (blue, solid line), training on data actively acquired by a CRF (green, dashed line) yields worse performance.

randomly choosing i.i.d. examples. This violates the standard supervised machine learning assumption, i.e., that the training and test data reflect the same underlying data distribution. However, AL has been found work well empirically with a variety of tasks and models [23, 20, 5, 28, 24].

A common intuition for why AL might be beneficial follows: Suppose that after training on 4000 examples, the learner is nearly 100% confident in the label of example i, but unsure of the label for example j. In this case, it seems likely that little additional information will be gained by observing the label for example i. By contrast, collecting an annotation on instance j seems likely to provide significantly more information to the learner, given its current uncertainty of j's label. We note that uncertainty-based AL is just one popular approach among many proposed heuristics for AL.

A key consideration in the design of AL algorithms is the scoring function. While uncertainty sampling [14, 21, 8] remains the dominant approach, how precisely to quantify model uncertainty remains a largely open question [9, 4], and thus a variety of heuristics have been proposed in the literature. One commonality among these heuristics is that they generally depend somehow on the underlying model being trained [22, 23] — we refer to this model as the *acquisition model*. Consequently, the collected training data and the acquisition model are *coupled*.

This coupling becomes problematic in light of a common pattern in practice: manually labeled data tends to have a longer shelf life than models, largely because it is expensive to acquire. At the same time, progress in machine learning is fast. In many settings, an actively acquired dataset may thus remain in use (much) longer than the source model used to acquire it. In these cases a few natural questions arise: How does a *successor* model S fare, when trained on data collected via an acquisition model A? How does this compared training S on natively acquired data? How does it compare to training S on i.i.d. data?

For example, if we use uncertainty sampling under a logistic regression model to acquire a training corpus $\mathcal{D}$, and subsequently train a Convolutional Neural Network (CNN) using $\mathcal{D}$, will the CNN perform better than it would have if trained on a training set acquired via i.i.d. random sampling? And how does it perform relative to using a training corpus actively acquired using the CNN as the acquisition model?

Figure 1 depicts results for a sequence tagging example, specifically Named Entity Recognition (NER) on the CoNLL 2003 corpus [25]. We consider two tagging models: a standard Conditional Random Field (CRF) [13] and the recently proposed neural BiLSTM-CNN model [24]. We observe that training the latter with a dataset actively acquired using the former yields predictive performance (in terms of F1) that is *worse* than that achieved under i.i.d. sampling. Given that datasets tend to outlast models,[2] these results raise questions regarding the benefits of using AL in practice. As far as we are aware, these questions—which have obvious practical implications—have not previously been explored empirically.

2 Measuring the Transferability of Actively-Acquired Data

To investigate the transferability of actively acquired datasets across models, we investigate two tasks in natural language processing for which AL has been previously shown to confer substantial benefits: text classification, and sequence tagging—namely NER.[3] Here we consider both linear models and those more representative of the current state-of-the-art for these tasks. We investigate each possible (acquisition, successor) pair among the considered models. For each pair (A, S), we first simulate iterative active data acquisition with model A to label a training dataset $\mathcal{D}_A$. We then train the successor model S using $\mathcal{D}_A$. In our evaluation, we compare the relative performance (accuracy or F1, as appropriate for the task) of the successor model trained with corpus $\mathcal{D}_A$ to the scores achieved by training on comparable amounts of native and i.i.d. data.

We simulate pool-based AL using labeled benchmark datasets by withholding document labels from the models. This induces a simulated pool of unlabeled data $\mathcal{U}$. In AL, it is common to *warm-start* the acquisition model, training on some amount of i.i.d. labeled data $\mathcal{D}_w$ before using the model to score candidates in $\mathcal{U}$ [22] and commencing the AL process. We follow this convention throughout.

Once we have trained the acquisition model on the warm-start data, we begin the simulated AL loop, iteratively selecting instances for labeling and adding them to the dataset. We denote the dataset acquired by model A at iteration t by $\mathcal{D}_A^t$; $\mathcal{D}_A^0$ is initialized to $\mathcal{D}_w$ for all models (i.e., all values of A). At each iteration, the acquisition model is trained on $\mathcal{D}_A^t$. It then scores the remaining unlabeled documents in $U \setminus \mathcal{D}_A^t$ according to a standard uncertainty AL heuristic. The top n candidates, $\mathcal{C}_A^t$, are selected for (simulated) annotation, their labels are revealed, and they are added to the training set: $\mathcal{D}_A^{t+1} \leftarrow \mathcal{D}_A^t \cup \mathcal{C}_A^t$. At the experiment's conclusion (time step T), each acquisition model A will have selected a (typically distinct) subset of $\mathcal{U}$ for training.

Once we have acquired datasets from each acquisition model $\mathcal{D}_A$, we evaluate the performance of each possible successor model when trained on $\mathcal{D}_A$. Specifically, we train each successor model S on the acquired data $\mathcal{D}_A^t$ for all t in the range $[0, T]$, evaluating its performance on a held-out test set (distinct from $\mathcal{U}$). We compare the performance achieved in this case to

[2]Indeed, the CoNLL NER dataset is a good example of this; this was collected in 2003 but remains in wide use.

[3]Recent works have shown that AL is effective for these tasks even when using modern, data-hungry neural architectures [28, 24], although these did not explore the question of transferability.

that obtained using an i.i.d. training set of the same size, and to the performance achieved using the successor's native actively acquired set, $\mathcal{D}_S^t$ at each t.

We run these experiments ten times, averaging the results to create summary learning curves, as shown in Figure 1. These quantify the comparative performance of a particular model achieved using the same amount of supervision, but elicited under different acquisition models. For each model, we compare the learning curves of each acquisition strategy, including active acquisition using a 'foreign' model and subsequent transfer, active acquisition without changing models (i.e., the typical AL case), and the baseline strategy of i.i.d. acquisition.

3 Tasks

We now briefly describe the models, datasets, acquisition functions, and implementation details for the experiments we conduct with active learners for sentence classification (3.1) and NER (3.2).

3.1 Text Classification

3.1.1 Models

We consider three standard models for text classification: Logistic Regression (LR), Convolutional Neural Networks (CNNs) [10, 27], and Long Short-Term Memory (LSTM) networks [6]. For LR, we represent texts via sparse, TF-IDF bag-of-words (BoW) vectors. For the neural models (CNN and LSTM), we represent each document as a sequence of word embeddings, stacked into an $l \times d$ matrix where l is the length of the sentence and d is the dimensionality of the word embeddings. We initialize all word embeddings with pre-trained word2vec vectors [16]. We initialize the vector representations for all words lacking pre-trained vectors uniformly at random. For the CNN, we impose a maximum sentence length of 120 words, truncating sentences exceeding this length and padding shorter sentences. We used filter sizes of 3, 4, and 5, with 128 filters per size. For LSTMs, we limited sentences to 40 words.[4] We trained all neural models using the Adam optimizer [11], with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_1 = 0.999$, and $\epsilon = 10^{-8}$.

For LR, CNN, and LSTM, we use the entropy variant of uncertainty sampling, which is perhaps the most widely used AL heuristic [22]. This strategy selects a document for annotation according to the function

$$\underset{\mathbf{x} \in \mathcal{U}}{\operatorname{argmax}} - \sum_j P(y_j|\mathbf{x}) \log P(y_j|\mathbf{x}),$$

where $\mathbf{x}$ are instances in the pool $\mathcal{U}$, j indexes potential labels of these (we have elided the instance index here) and $P(y_j|\mathbf{x})$ is the predicted probability that $\mathbf{x}$ belongs to class y_j (this estimate is implicitly conditioned on a model that can provide such estimates).

3.1.2 Datasets

We perform text classification experiments using four benchmark datasets. We reserve 20% of each dataset (sampled at i.i.d. random) as test data, and use the remaining 80% as the pool of unlabeled data $\mathcal{U}$. We sample 100 documents randomly from $\mathcal{U}$ as $\mathcal{D}_w$. All models receive the same $\mathcal{D}_w$ for any given run.

[4] Passing longer sentences to the LSTM degraded performance in preliminary experiments.

Table 1: Text classification dataset statistics.

Dataset	# Classes	# Documents	Examples per Class
Movie Reviews	2	10662	5331, 5331
Subjectivity	2	10000	5000, 5000
TREC	6	5952	1300, 916, 95, 1288, 1344, 1009
Customer Reviews	2	3775	1368, 2407

- **Movie Reviews**: This corpus consists of sentences drawn from movie reviews. The task is to classify sentences as expressing either positive or negative sentiment [18].
- **Subjectivity**: This dataset consists of objective and subjective statements and the task is to classify them accordingly [17].
- **TREC**: This task requires the learner to categorize questions into one of six categories [15] based on the subject of the question (e.g., questions about people, locations, and so on). The TREC dataset defines standard train/test splits. However, we generate a different split for consistency, as the suggested split does not follow the ratio we used.
- **Customer Reviews**: This set is composed of reviews of various products, and the task is to categorize these as positive or negative [7].

3.2 Named Entity Recognition

3.2.1 Models

We consider transfer between two NER models: Conditional Random Fields (CRF) [13] and Bidirectional LSTM-CNNs (BiLSTM-CNNs) [2].

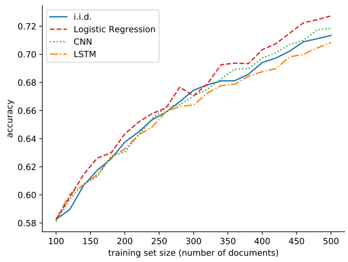
To train the CRF, we use models with features for each word including word-level and character-based embeddings, word suffix, capitalization, digit contents, and part-of-speech tags. The BiLSTM-CNN model[5] initializes word vectors to pre-trained GloVe vector embeddings [19]. We learn all word and character level features from scratch initializing with random embeddings.

For both CRF and BiLSTM-CNN, we use maximized normalized log-probability (MNLP) [24] as our AL heuristic. MNLP adapts the least confidence heuristics to sequences. To avoid favoring selecting longer sentences (owing to the lower probability of getting the entire tag sequence right), we normalize the log probabilities of the predicted tag sequence by the sequence length and sort in ascending order according to the function

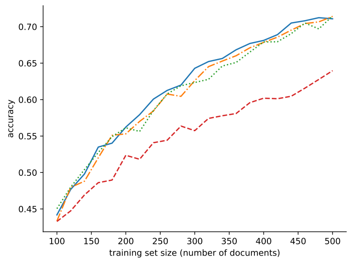
$$\max_{y_1,\ldots,y_n} \frac{1}{n} \sum_{j=1}^{n} \log P(y_i | y_1, \ldots, y_{n-1}, \mathbf{x})$$

Where the max over y assignments denotes the most likely set of tags for instance $\mathbf{x}$ and n is the sequence length. Because explicitly calculating the most likely tag sequence is computationally expensive, we follow [24] in using a greedy decoding (i.e., beam search with width 1) to determine the model's prediction.

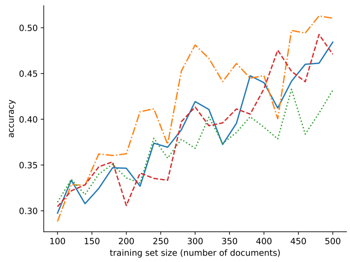
[5] Implementation of the BiLSTM-CNN is available at https://github.com/asiddhant/Active-NLP.



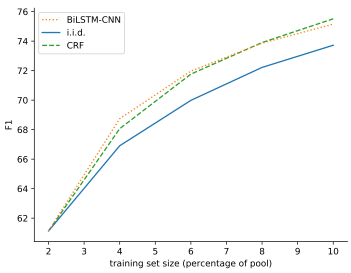
(a) LR on TREC dataset



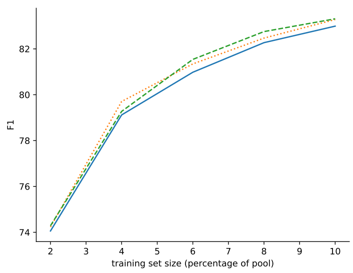
(b) CNN on TREC dataset



(c) LSTM on TREC dataset



(d) CRF on OntoNotes dataset



(e) BiLSTM-CNN on OntoNotes dataset

Figure 2: Sample learning curves for the text classification task on the TREC dataset and the NER task on the OntoNotes dataset (we report learning curves for all models and datasets in the Appendix). Individual plots correspond to successor models. Each line corresponds to an acquisition model, with the blue line representing an i.i.d. baseline.

3.3 Datasets

We perform NER experiments on the CoNLL-2003 and OntoNotes-5.0 English datasets. We used the standard test sets for both corpora, and merged the standard training and validation sets to form $\mathcal{U}$. Because the size of these datasets differ significantly (Table 3), we initialize $\mathcal{D}_{seed}$ to 2% of $\mathcal{U}$.

- **CoNLL-2003**: This dataset consists of sentences drawn from Reuters news, with individual words tagged as person, location, organization, or miscellaneous entities using an IOB scheme [25]. The corpus contains 301,418 words.
- **OntoNotes-5.0**: A corpus of sentences drawn from a wide variety of sources and genres, including newswire, broadcast news, broadcast conversation, and web data. Words are categorized using eighteen entity categories annotated in IOB scheme [26]. The corpus contains 2,053,446 words.

4 Results

We compare transfer between all possible (acquisition, successor) model pairs for each task. We demonstrate the performance of each model under transfer both through tables of quantitative results (Table 2 and Table 3 for classification and NER, respectively) and graphically by plotting learning curves of performance vs training set size (Figure 2). We report additional results, including all learning curves (for all model pairs and for all tasks), in the Appendix. For the text classification task, we use accuracy as our performance metric. For the NER task, we use F1.

Table 2: Text classification accuracy, evaluated for each combination of acquisition and successor models with training sets composed of 300 and 500 documents. Colors indicate performance relative to i.i.d. baselines: Blue implies that a model fared better, red that it performed worse, and black that it performed the same.

Text Classification

	Acquisition Model							
	300 Documents				500 Documents			
Successor Model	i.i.d.	LR	CNN	LSTM	i.i.d.	LR	CNN	LSTM
Movie Reviews								
LR	61.7	**61.9**	61.0	61.2	64.0	**64.3**	63.4	63.3
CNN	70.3	69.8	**69.4**	68.7	72.6	72.6	**72.0**	70.6
LSTM	61.4	62.3	61.6	**61.4**	67.9	69.2	67.6	**65.5**
Subjectivity								
LR	82.4	**83.0**	81.7	80.9	84.2	**85.3**	83.0	82.1
CNN	87.3	86.6	**86.9**	86.8	88.9	88.2	**89.0**	88.3
LSTM	83.4	83.0	83.2	**84.4**	87.6	86.8	87.5	**88.0**
TREC								
LR	67.4	**67.1**	67.0	66.4	71.3	**72.7**	71.9	70.8
CNN	64.3	55.7	**62.3**	62.6	71.1	64.0	**71.5**	71.4
LSTM	41.9	41.3	36.8	**48.1**	48.4	47.1	43.2	**51.0**
Customer Reviews								
LR	71.9	**71.0**	69.4	71.2	73.2	**73.3**	71.4	72.9
CNN	70.8	72.2	**72.4**	72.3	74.6	75.6	**74.3**	75.8
LSTM	64.6	66.2	61.0	**66.3**	70.6	68.4	67.5	**71.8**

To compare the learning curves, we select incremental points along the x-axis and report the performance at these points. This allows direct comparison of results. For text classification, we report the performance achieved using training sets containing 300 and 500 documents. These are presented in Table 2. For the NER task, we report F1 with training sets containing 6% and 10% of the pool (Table 3).

5 Discussion

Results in Tables 2 and 3 confirm that standard AL — when the acquisition and successor models are one and the same — tends to outperform i.i.d. sampling more often than not. This is consistent with a large body of prior work both with linear models [23, 20], and more recent work on deep active learning for NLP [28, 24]. However, our results suggest that models trained on *foreign* actively acquired datasets tend to underperform i.i.d. datasets. We observe the clearly in the text classification task, where only a handful of (acquisition, successor) pairs lead to performance greater than that attained by i.i.d. data. We observe performance greater than the i.i.d. baseline in only 21% of the tabulated data points representing dataset transfer (in which acquisition and successor models differ). This is in contrast to 63% of tabulated data points representing standard AL.

Table 3: F1 measurements for the named entity recognition task, taken with training sets composed of 6% and 10% of the training pool.

Named Entity Recognition

	Acquisition Model					
	6% of Pool			10% of Pool		
Successor Model	i.i.d.	CRF	BiLSTM-CNN	i.i.d.	CRF	BiLSTM-CNN
CoNLL						
CRF	65.6	**66.8**	66.9	69.3	**70.0**	70.2
BiLSTM-CNN	85.2	85.0	**86.0**	87.8	87.1	**87.9**
OntoNotes						
CRF	70.0	**71.8**	72.0	73.7	**75.5**	75.2
BiLSTM-CNN	81.0	81.5	**81.3**	83.0	83.3	**83.3**

The only (acquisition, successor) pairs that exhibit greater than random performance at both sample points are LR to LSTM with the Movie Reviews corpus, and LR or LSTM to CNN with the Customer Reviews corpus. We do see greater than random (i.i.d.) performance at 300 under transfer from CNN to LSTM with the Movie Reviews corpus, as well as LR to LSTM with the customer review corpus. However, these benefits disappear by the time we reach 500 documents. At this point, we start to observe beneficial transfer from CNN to LR and LSTM to CNN with the TREC dataset.

Notably, we see no consistent pattern to when transfer is beneficial. In the findings reported here, such cases are relatively evenly distributed across datasets, acquisition models, and successor models. The only (acquisition, successor) pairs to produce a beneficial outcome on more than one dataset are LR to LSTM and LSTM to CNN. Even for these pairs, we see beneficial outcomes at less than half of our sample points. Given this observation, it may be difficult to predict whether transfer from an acquisition model to a successor model given a particular dataset will be beneficial or harmful (as compared to i.i.d. sampling) without actually performing AL and dataset transfer. This is clearly not practicable in a real-world setting.

The results for NER are somewhat more favorable for AL, where a higher percentage of pairs yield improved performance versus an i.i.d. baseline. Specifically, only CRF to BiLSTM-CNN performs worse than i.i.d., and then only on the CoNLL dataset. For this task, we see performance above the i.i.d. baseline in 75% of transfer data points, compared to 100% of standard AL data points.

6 Conclusion

In this paper, we have examined the results of transferring an actively sampled training data set from an acquisition model (used to acquire the data) to a distinct successor model. Given the longevity and value of training sets and the frequency at which new machine learning models advance the state-of-the-art, this should be an anticipated scenario in any active learning regime, i.e., annotated data often outlives models. We have analyzed the results of such transfer via an empirical study including two standard natural language processing tasks, coupled with multiple datasets, and acquisition, successor model pairs with each.

Our findings indicate that transferred actively learned training sets often result in performance worse than that attained using an equivalently sized i.i.d. training set, especially in the case of text classification. There are specific combinations of machine learning task, dataset, and (acquisition, successor) pairs of models for which transfer produces better-than-i.i.d. results. However, it is difficult to predict beforehand whether any particular combination will produce such benefit. The only factor that seems to provide significant predictive power is the machine learning task, and even this is not completely reliable. These findings suggest that AL should be used with caution in situations where one hopes to re-use the actively acquired data to train novel models in the future.

References

[1] Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, 2017.

[2] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*, 2015.

[3] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

[4] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning*, 2016.

[5] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192, 2017.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[7] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[8] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[9] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, 2017.

[10] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012.

[13] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[14] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

[15] Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositi pdf.

[17] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, 2004.

[18] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, 2005.

[19] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

[20] Maria E. Ramirez-Loaiza, Manali Sharma, Geet Kumar, and Mustafa Bilgic. Active learning: an empirical study of common baselines. *Data Mining and Knowledge Discovery*, 31(2):287–313, Mar 2017. ISSN 1573-756X. doi: 10.1007/s10618-016-0469-7. URL https://doi.org/10.1007/s10618-016-0469-7.

[21] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML*, pages 441–448, 2001.

[22] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[23] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[24] Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ry018WZAZ.

[25] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada, 2003.

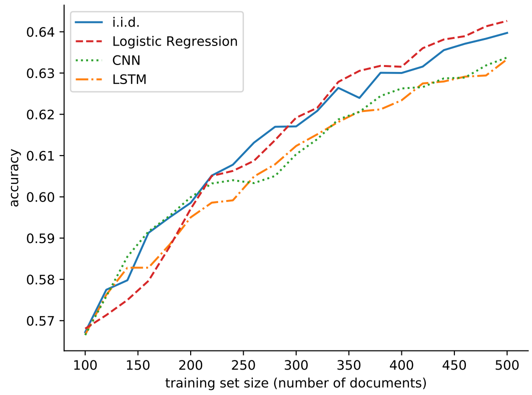
[26] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 2013.

[27] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

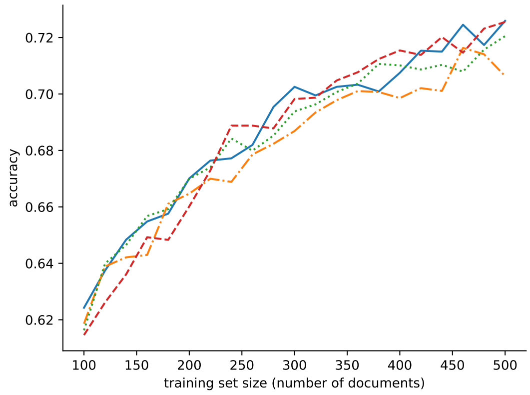
[28] Ye Zhang, Matthew Lease, and Byron C Wallace. Active discriminative text representation learning. In *AAAI*, 2017.

Appendix

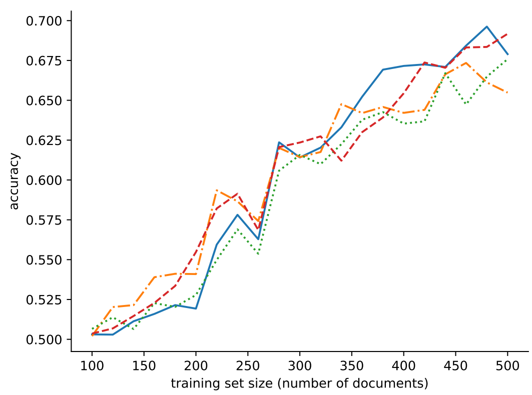
Figure 3: This appendix contains the full set of collected learning curves for both the text classification task and the NER task.



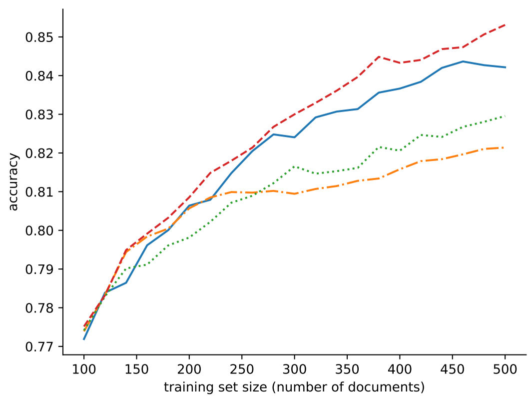
(a) LR on Movie Reviews dataset



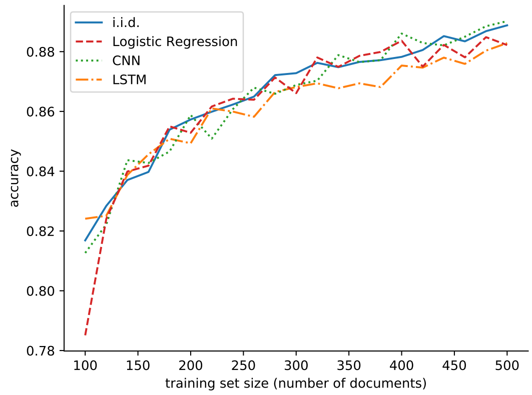
(b) CNN on Movie Reviews dataset



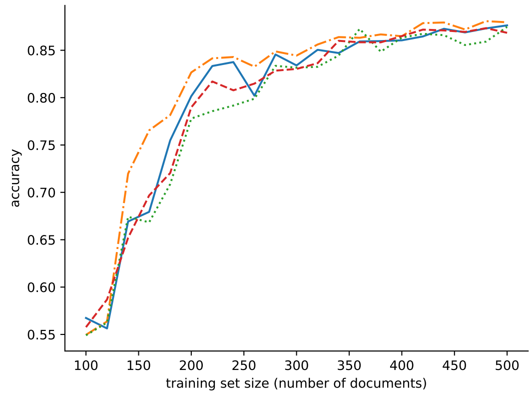
(c) LSTM on Movie Reviews dataset



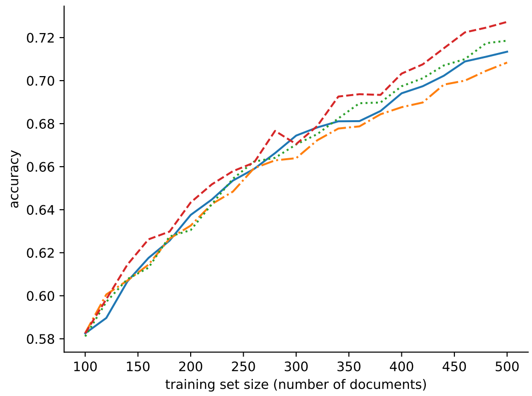
(d) LR on Subjectivity dataset



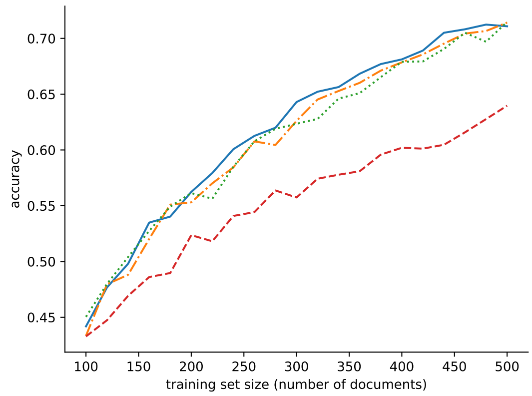
(e) CNN on Subjectivity dataset



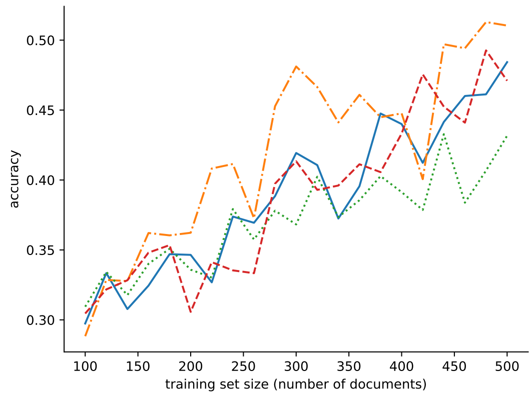
(f) LSTM on Subjectivity dataset



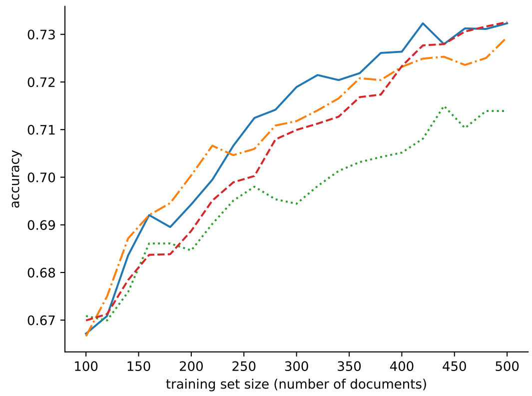
(g) LR on TREC dataset



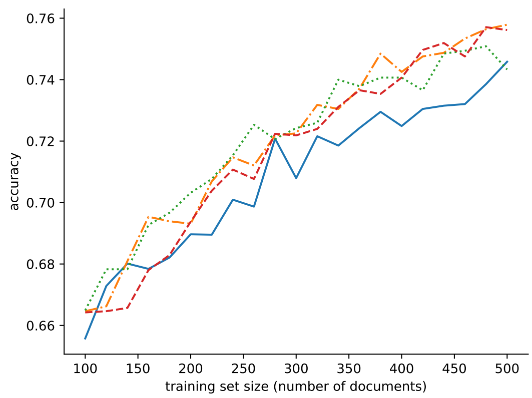
(h) CNN on TREC dataset



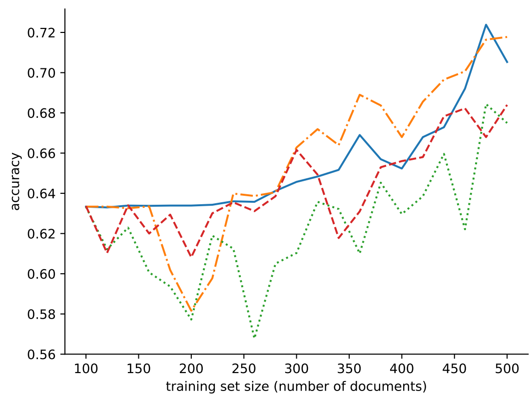
(i) LSTM on TREC dataset



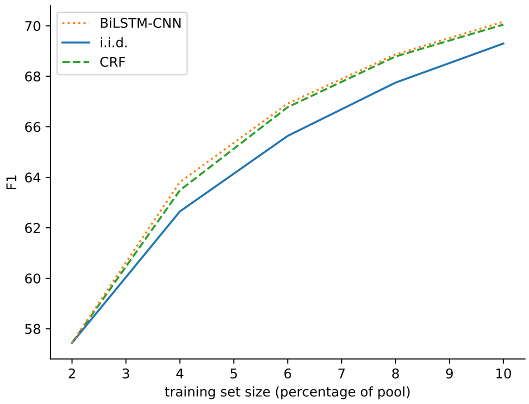
(j) LR on Customer Review dataset



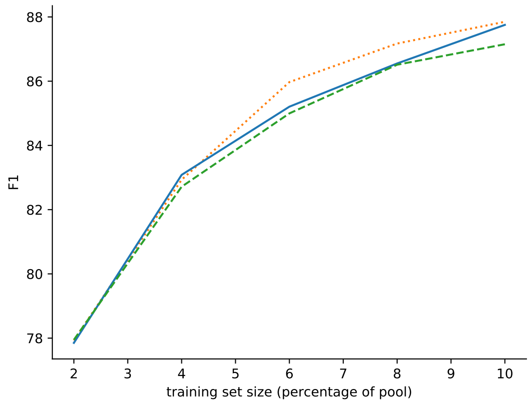
(k) CNN on Customer Review dataset



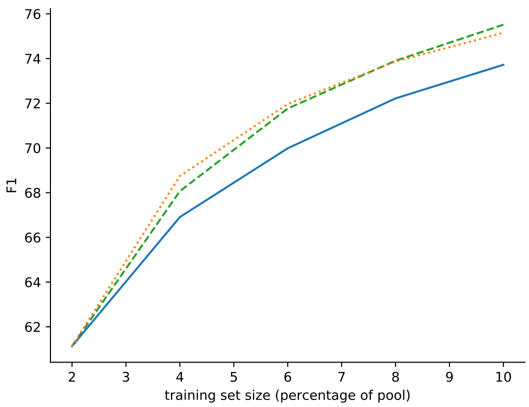
(l) LSTM on Customer Review dataset



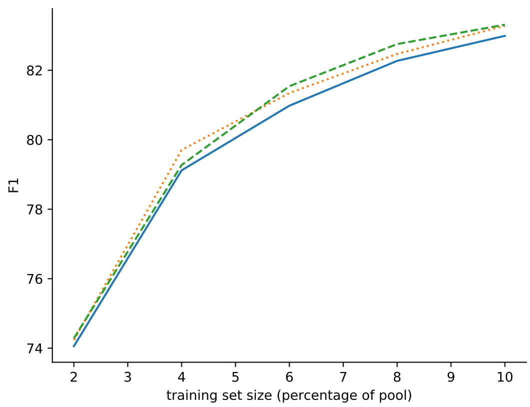
(m) CRF on CoNLL dataset



(n) BiLSTM-CNN on CoNLL dataset



(o) CRF on OntoNotes dataset



(p) BiLSTM-CNN on OntoNotes dataset